

International Journal of Software Engineering and Computer Systems (IJSECS)

ISSN: 2289-8522, Volume 3, pp. 106-119, February 2017

©Universiti Malaysia Pahang

DOI: <http://dx.doi.org/10.15282/ijsecs.3.2017.7.0029>

FAULT TOLERANCE FOR TWO WHEEL MOBILE ROBOT USING FSM (FINITE STATE MACHINE)

Chan Shi Jing¹, Dwi Pebrianti¹, Mohammad Fadhil Abas¹, Luhur Bayuaji²

¹Faculty of Electrical & Electronics Engineering,
University Malaysia Pahang Campus Pekan, 26600 Pekan,
Pahang, Malaysia.

²Faculty of Computer Science and Software Engineering,
University Malaysia Pahang, 26600 Pekan,
Pahang Darul Makmur, Malaysia.

e-mail: chanshijing@gmail.com, dwipebrianti@ump.edu.my, mfadhil@ump.edu.my,
luhurbayuaji@ump.edu.my

ABSTRACT

Fault Tolerance (FT) enables system to continue operating despite in the event of failures. Therefore, FT serves as a backup component or procedure that can immediately play its role to minimize any service lost. FT exists in many forms, where it can either be in the software form or hardware form or both hardware and software form. Fault Tolerance is an umbrella term for fault detection, fault isolation, fault identification and fault solving. To better visualize the fault detection and isolation process, a two wheel robot is used in this study to represent the complex system. The aim of this research is to construct and design a Fault Tolerance algorithm considered to speed up the fault isolation procedure and it might identify multiple fault with the same static fault signature. The Finite State Machine (FSM) model, a wide library of reusable model for the fault tolerant is used in this study to solve the fault in actuator or in the sensor by resetting and adjusting it to the correct position. Using the system sensors or actuators, the technique used is able to recognize the fault from its data. This FSM method is capable to avoid, replace, reset and recover any possible faults occurred in the system, offering an innovative solution to identify and solve a fault immediately.

Keywords: Kalman filter; Artificial Neural Network; Fault tolerant; Fault detection; Fault Isolation

INTRODUCTION

Fault tolerance (FT) is a subsystem of the control system, which serves as a condition solution to solve the fault or failure (hardware or software) in the system. Most researches focused on how to identify the system fault, including failure of the actuator and sensor as well as the error of the system. Researchers have proposed various solutions to the problem including (Magni, Scattolini, & Rossi, 2000) the Finite State Machine (FSM) method to identify the type of fault, (Apley, Shi, & Arbor, n.d., 1998) the GLRT mathematical method to identify the fault, (Bouhouche, Lahreche, Ziani, & Bast, 2005) artificial neural network, (Lee, Alena, & Robinson, 2005) fault decision method, (Bai, Dick, Dinda, & Chou, 2011) Fault-Aware Code Transformation

For Sensor Network (FACT), (Mathew et al., 2014) genetic algorithm, and (Kamel, Yu, & Zhang, 2016) two-stage Kalman filter. On the other hand, computation methods proposed to solve this problem are (Benowitz, Calhoun, & Alderson, 1975) the Advanced Avionics Fault Isolation System (AAFIS) concept utilizes bit (build in test) and etc. Most of solutions focused on identifying the fault of the system and will stop upon fault identification. The fault that has been identified would require human to take action to fix the system.

The most frequent FT solution is the Petri nets which were used in rollback recovery with check pointing, recovery blocks, N-version programming, and conversations (Shieh, Ghosal, Chintamaneni, & Tripathi, 1989). The Petri nets are also known as a fault-tolerant parallel processor, (Kanoun, Kaaniche, Baoune, Laprie, & Arlat, 1993) a Markov chain of the fault-tolerant software system. This algorithm automatically constructs the Petri net models using parameterized subnet primitives. The numerical results clearly illustrated the applicability of the proposed models, where (Dugan, Bavuso, & Boyd, 1992) dynamic fault-tree modeling techniques is capable of handling high levels of reliability, frequently employ high levels of redundancy, dynamic redundancy management, and complex fault & error recovery techniques. However, this type of FT system is not suitable in several field of application. Based on the problem stated, the solution proposed in this study is to design an algorithm which is able to generate self-identification and self-correction on system faults.

In this paper, a self-fix FT algorithm is proposed to solve the fault in a Two Wheel Mobile Robot (TWMR). It should be noted that the algorithm can only work well in the happening of non-severity failure. This TWMR mimics the working principle of ground vehicles such as car and bus. Therefore, this study will enhance the safety measures in transportation industry (Dedeoglu, Roumeliotis, & Sukhatme, 2000). Furthermore, the result of this study can be used to further improve the performance of complex system. Once the decision making process of the FT system obtain a more precise value of fault parameter, it will undergo faster processing time that will facilitate the identification of the failure position as well as the severity of failure precisely.

This paper will focus on a very simple technique, which transform the FSM from an observer based technique towards a solution based technique. The modified FSM subsystem is capable to order or instruct the system to work under the possible fault in the system. The state FSM acts as an observer, which is capable of detecting specific type of faults in minimum time (Dedeoglu et al., 2000). The FSM method is commonly used in the system communication including, (Zhu et al., 2015) Independent observer to detect fault in communication system as well as (Foullas, Karras, & Kyriakopoulos, 2015) multiple observers into the one of fault tolerant to identify if it is capable of detecting the execution error of an ISO transport protocol even when a subset of the observers is faulty.

The next section will review the concept of FSM and explain how FSM model serves as the solution solving methodology. The model is able to indicate the desired target to decompose the observer fault class from the Artificial Neural Network (ANN) into a set of smaller FSM solution that is capable of jointly solving faults. This paper will also provide a virtual alternate procedure which is capable of operating in real time that eliminates synchronization problems arising in the original method.

MODEL APPROACH

The research objective is to develop an algorithm that is capable to solve the fault system automatically during their operation life, at the same time considering the performance of the system (Wang & Schwartz, 1992). Therefore, only relevant result for a given period of time is obtained to ensure stable and reliable software condition. This section will briefly describe the various fault-tolerant techniques used.

There are several condition faults tolerant approvals used in this paper to prevent non-responding error in the system. The four main types of responding approaches taken are:

- i. *Recovery* type: to encapsulate each critical function of a process state (Foullas et al., 2015). In the event of a failure, a checkpoint allows the robot to reverse its motion to the previous condition. Therefore, stored checkpoints of a process are same nodes where the executed process can be conducted to recover from the failures.
- ii. *Reset* type: this scheme is activated when there is a critical functional fault in the system and it is unable to be solved by the other approach.
- iii. *Replace* type: to replace the available sensor that still working well in the robot for restoring the system function.
- iv. *Avoid* type: to avoid the fault that may be happened in the actuator by resetting the actuator. The following section will illustrate how the voiding type failure happens in the complex system.

Our approach of modeling the reliability of the fault tolerant software is more general as it combines information on the software structure as well as the reliability growth of the components. An important property of this FSM model is the transformation from a mathematical model the finite sequential loop to a stable and reliable FSM system which is very similar to FSM process. This is particularly relevant as it allows the reliability growth of fault tolerant software system to be modeled from the reliability growth of its components which allows the transformation based on the interpretation of the FSM model. Based on our research, this approach is among the easiest modeling of fault tolerant software system due to its systematic nature as electronic FSM methods.

METHODOLOGY

Firstly, the residual information was generated by comparing the value of the actual and the filtering value. This information will then be fed into an Artificial Neural Network (ANN) to identify the type of fault that might exist. If fault is detected, the system will do further adjustments using the FSM solution solving technique.

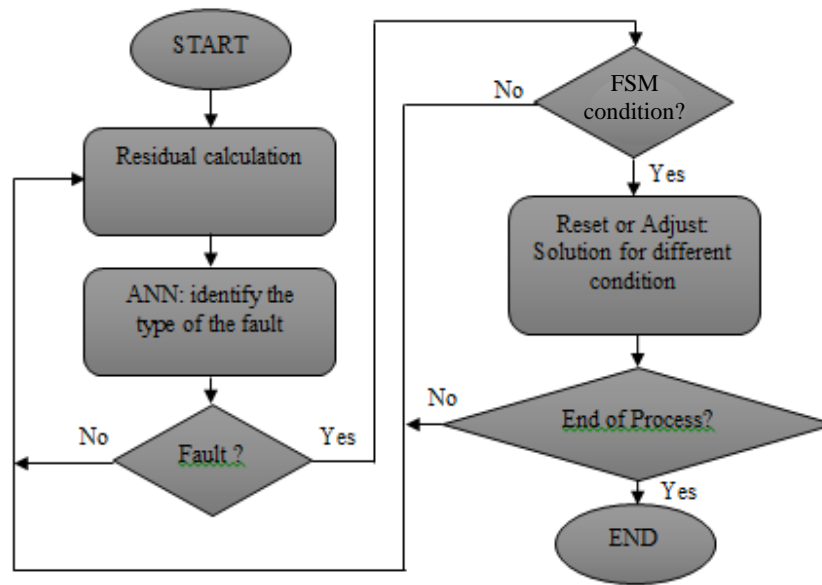


Figure 1. Flow planning of FT

This FT algorithm detects potentially faulty sensor or actuator in the TWMR and automatically generates solutions for each faulty condition. During the faulty of the TWMR, the type of fault behavior generated will be compared with the solutions that have been set or stated in the system, giving instruction to the robot to adjust its angle to reach the destination.

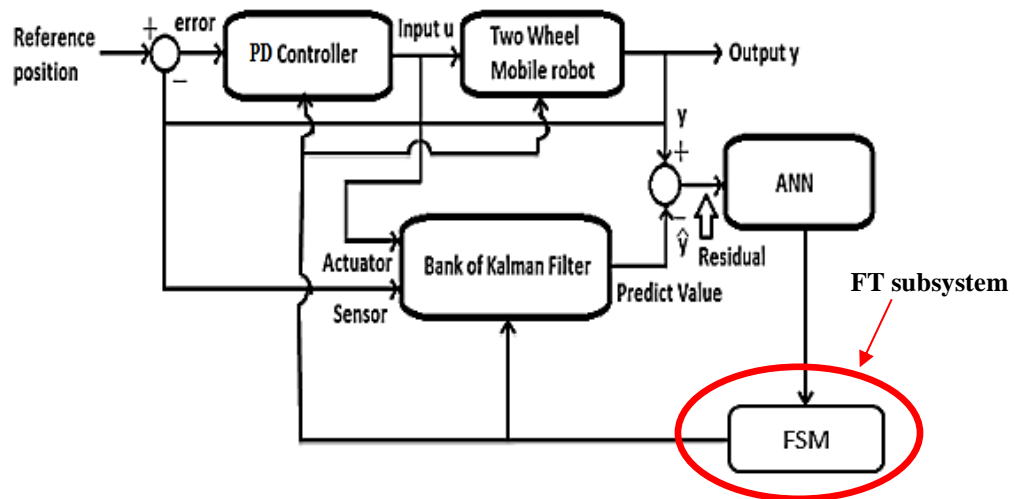


Figure 2. Block Diagram of the Complete System

In this system, the reference position is set so that TWMR will move to the reference position. This reference good position is compared with the position obtained from the system. The difference between these values is called error signal. This signal will be fed to a PD controller to generate the control input u to the system. The Kalman filter is used as an observer to predict the output value of the sensors and the actuators.

The result will then be compared to the system output y to generate the residuals. The purpose of using ANN in this FT subsystem is to convert the analog signals from the residual to discrete signal 0 and 1. This type of signal can then be easily used in fault classification task using the FSM subsystem.

FSM SUBSYSTEM

The basic principal in the identification of the state condition format is to decompose the studied FSM method (from the previous state to the next state). State is referred as the condition or the solution of a condition. Using this technique, we will be able to see the state condition in the system. In order to fulfill the condition before proceeding to the next state, the internal link of the system need to be taken care of as it is very important to the system. The internal link in the system is defined as the fault occurrence based on the observer result. The system is observed over a time interval and the relevant solution case generated by the system is recorded. A set of the relevant solution is treated and prepared by the system.

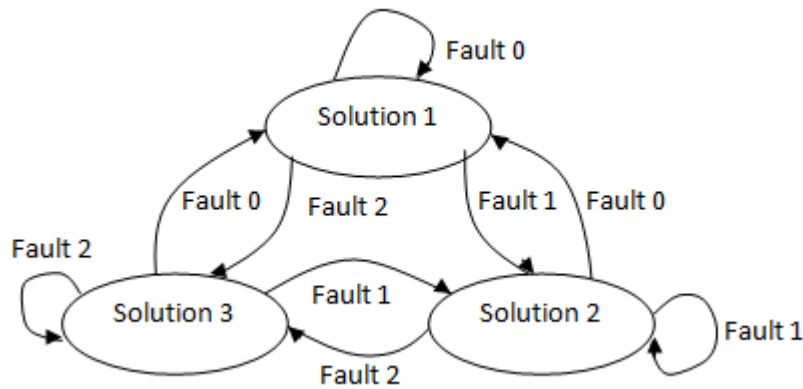


Figure 3. Finite State Machine solution

Technically, this assumption is useful for limited size of the FSM produced by the composition of the basic FSM model, although the process described below may be extended to relax this assumption. *L. Magni, M. Ravara, C. Rossi, and R. Scattolini* described a composition rule which allowed modification of two faults simultaneously (Magni et al., 2000). In addition, once failure is detected, the system is typically reconfigured to maintain safety standards. In the FSM configuration, any other fault must be considered a non-simultaneous failure of the security configuration. For a better representation, we are using the table format to derivate the FSM subsystem.

The model of the system is shown below.

The value 1 inside the box table is responding as Yes condition where fault occurred, while value 0 represent no fault in the system.

The solution for the system is stated in the form of FSM. The row represents i of the Fault occurred in the system corresponds to the solution state of faults occurred while the columns j represents the fault that might happen in the system.

The number of the resulting state FSM is depending on the number of the size on the probability of the combination faulty that might be occurred in the system. The computation complexity of the error solution finding is considerably reduced. (Lee et al., 2005)

Table 1. FSM Solution and Fault

Solution for the system	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5	Fault 6
S0	0	0	0	0	0	0
S1	1	0	0	0	0	0
S2	0	1	0	0	0	0
S3	0	0	1	0	0	0
S4	0	0	1	0	1	0
S5	0	0	1	1	1	0
S6	0	0	1	1	1	1
S7	1	1	1	1	1	1

In order to find the number of combination, we recall the basic of the FSM algorithm for N states (number of fault), which $N = 2^{m-1}$, when N is not a power of 2, a probability of the fault m combination number in order to make a power of 2. The decomposition algorithm can be generated reversely by $m = \log_2 N$. The elements of state s_N are the order $\{s_1, \dots, s_{n+1}\}$. The algorithm generates partitions that are symmetric.

Algorithm 1 (only 2 faults can be detected)

$$s_N = \{s_1, \dots, s_{n+1}\}; \quad s_N = \{s_1\} \quad (1)$$

For $N = 2^{m-1}$

A formal composition rule of FSM model is now derived under the following main signature:

Signature 1: the faults can occur in any second in the system while its diagnostic procedure is completed running.

Signature 2: the faults can occur or replace instantly in any second once a fault has solved, the diagnostic procedure is completed.

Technically, this signature is useful to limit the dimension of the FSM generated by the composition of elementary FSM model. Once a fault has been detected, the system is usually reconfigured in the state solution given in the system to maintain in safety standards. In the new system configuration, any further fault has to be viewed as non-simultaneous fault occurring to a safe configuration.

RESET AND ADJUST PROCEDURE

In practical, several faults might occur in the TWMR and these faults are categorized into three concern categories: the environment, the autonomous system and its environment awareness including the Exogenous event fault, Hardware fault, Software fault, Knowledge fault, Execution fault, and Sensing fault (Crestani & Godary-Dejean, 2012). Upon the in depth analysis of fault, it was found out that the acceleration impact on TWMR robots was due to friction between the wheels and the ground, sliding and brushless motoring of the wheels. Many case studies have shown that this is the most common factor leading to actuator failures, the loss of actuator power loss and control efficiency (Baghernezhad & Khorasani, 2013). This can subsequently lead to several problems with the mobile robot, preventing it from reaching the set point (position).

The FSM solution condition proposed in this research is a sequential looped solution state of the system. This solution state is passed through by the fault class that is assigned by ANN. It is assumed that the higher the types of solution condition in the system has the higher accuracy of the efficient work done by the robot correctness.

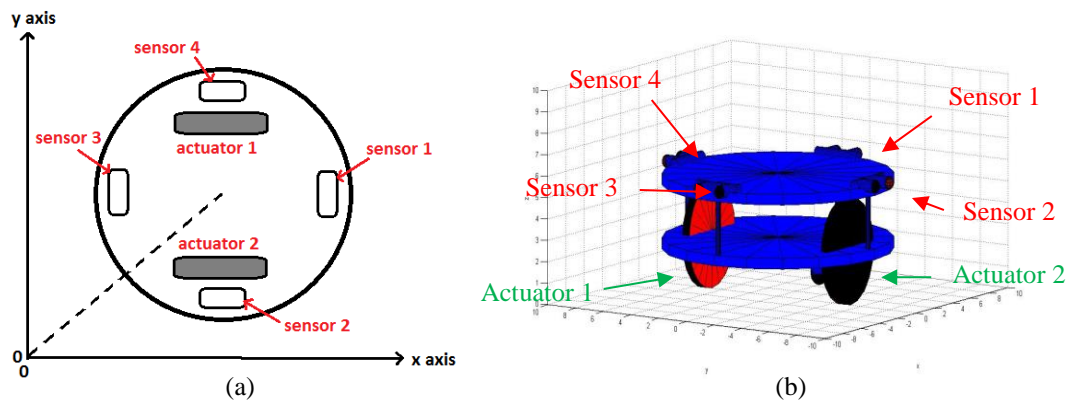


Figure 4. Schematic model of Two Wheel Mobile Robot

Figure 4 (a) shows that the schematic top view of the robot, (b) the 3D views of Two Wheel Mobile Robot. The method to develop the solution is based on each faulty condition:

Solution 0: The forward condition, running in the no fault conditions the actuator 1 and actuator 2.

Solution 1: The changes in speed or the turning of the actuator have been extensively studied to avoid the non-linear moving of the robot, running during the actuator 1 is in fault condition.. (Avoid type)

Solution 2: The changes in speed or the turning of the actuator have been extensively studied to avoid the non-linear moving of the robot, running during the actuator 2 is in fault condition. (Avoid type)

Solution 3: The reverse condition, running in the fault of the sensor 1 by replacing the sensor 3 as the front side detect distance sensor. (Replace type)

Solution 4: No detection working in the moving condition and depend on the distance that is detected by the sensor 2, running in the fault of the sensor 1 and

sensor 3 by replacing it with sensor 2 to detect the front side distance condition. Detecting in the time every moving 1 to 1 cm. (Replace type)

Solution 5: No detection working in the moving condition and depend on the distance that detected by the sensor 4, running in the fault of the sensor 1, sensor 3 and sensor 2 by replacing it with sensor 4 to detect the front side distance condition. Detecting time every moving 1 to 1 cm. (Replace type)

Solution 6: Free sensor mode to ensure the actuator 1 and actuator 2 are running properly. (Reset type)

Solution 7: critical condition stops working for the system. (Recover type)

In this section, we have considered eight types of solution to solve the six type of possible fault that might occur in the TWMR. In the next section, we will describe the scenario of the voiding method fault elimination method that has been applied to the actuator where there is 10 percent of the fault overshoot error on the system.

RESULT AND DISCUSSION

To illustrate the functionality of the FT subsystem with the TWMR and the solvability of the system with used FSM technique, three cases have been simulated using both algorithms. The parameter of the ANN was determined to achieve a balanced ratio between the detectability and false-alarm rates, based on the calculated integration value. This calculated integration is generated by a set of input and output data that trained with back-propagation in the learning phase. The failure on the actuator and sensor were injected at the various moments during the maneuvers. The parameters of the FSM scheme have been selected based on a limited number of cases. Then, this FSM scheme will show a better solution for those cases occurred in the TWMR.

A. Simulation 1

In this study, an actuator fault is implemented abruptly after the 4th second of the simulation time by 10 percent to the actuator 1. This fault can be due to deformations, flatten of the tires, broken, etc. The fault voiding method in the system for the failure that occurs in time 4s is used. The voiding method is normally used in the actuator fault condition. This is because there is no replacement actuator available in the TWMR. In addition, the percentage and the frequency of the fault occurrence in the TWMR were also being considered.

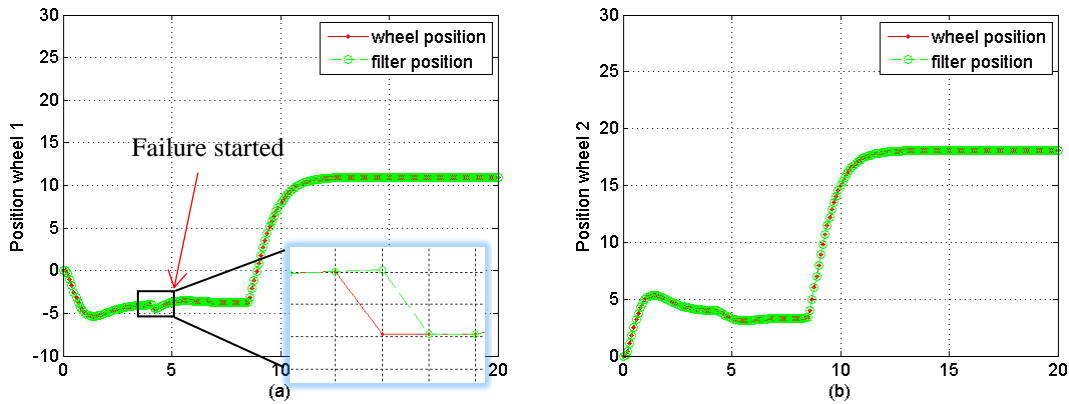


Figure 5. The Actuator 1 Fault in 4th Second

The entire actuator faults have been tested. At each time t the result of hypotheses will update the status each time. The fault appears on t time moment $t = 4s$ and the out u_t of the sensor is 10% up of its nominal value. Figure 5 represents both the real position (dot line) and the predicted position (circle line).

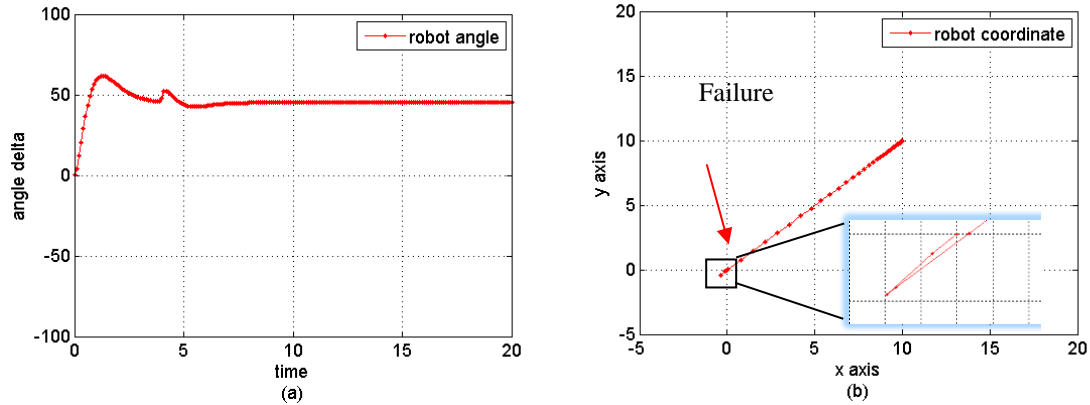


Figure 6. The Angle and Map of The Two Wheel Mobile Robot

Figure 6 (a) shows that the angle rotation of the TWMR, angle against time graphs in first 20 second. Figure 6 (b) shows the location of the robot in every 0.1 second, and the initial value started from (0, 0) and end with (10, 10).

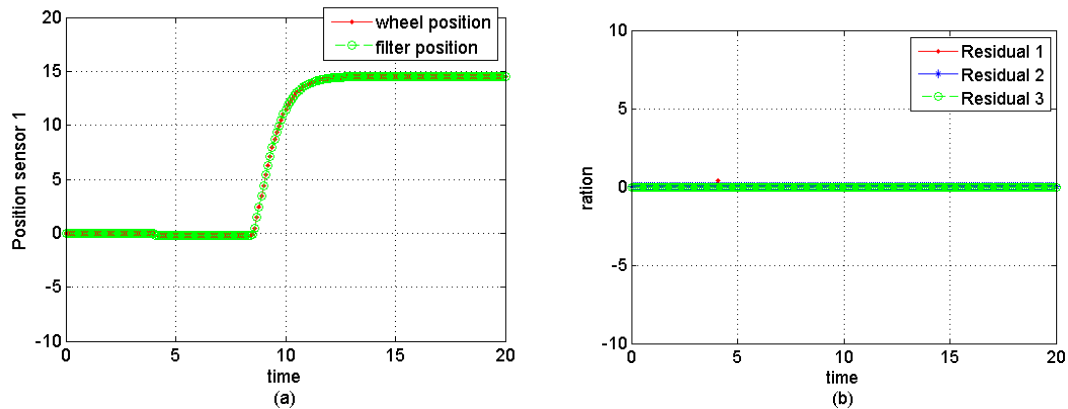


Figure 7. Sensor 1 Position Signal and Residual of the Two Wheel Mobile Robot

Figure 7 (a) shows the front sensor range distance against time in first 10 second. Figure 7 (b) shows the three different types of the resident that have been generated ($Re1$, $Re2$, and ($Re3/ Re5$) ($Re4$, $Re6$ can be ignored in this condition)). This residual are generated from comparing the sensor and actuator output with the estimated value of the Kalman Filter.

B. Simulation 2

The second case presented the happening of unflatten grounding while TWMR is running to the reference good position. This fault was included in the actuator fault and the stabilization failure was added after the 7th second of the simulation time by 10 percent to the actuator 2. The fault voiding method (solution 3) is used when the

actuator 2 is in fault condition. The changes in speed or the turning of the actuator have been extensively studied to avoid non-linear moving of the robot.

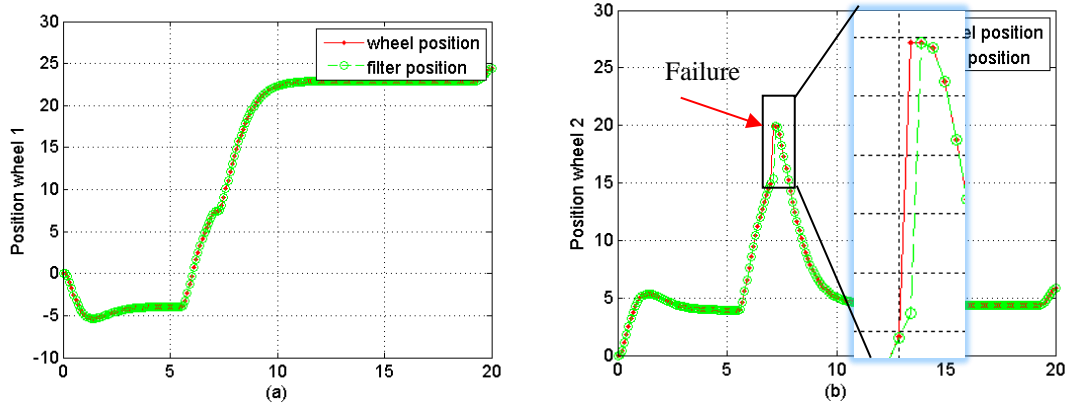


Figure 8. The Actuator 2 Fault in 7th Second

The actuator faults have been tested and apply in the TWMR system. At each time t the result of hypotheses will update the status each time. The fault appears on t time moment $t = 7s$ and the out u_t of the sensor is 10% up of its nominal value. Figure 8 represents both the real position (dot line) and the predicted position (circle line).

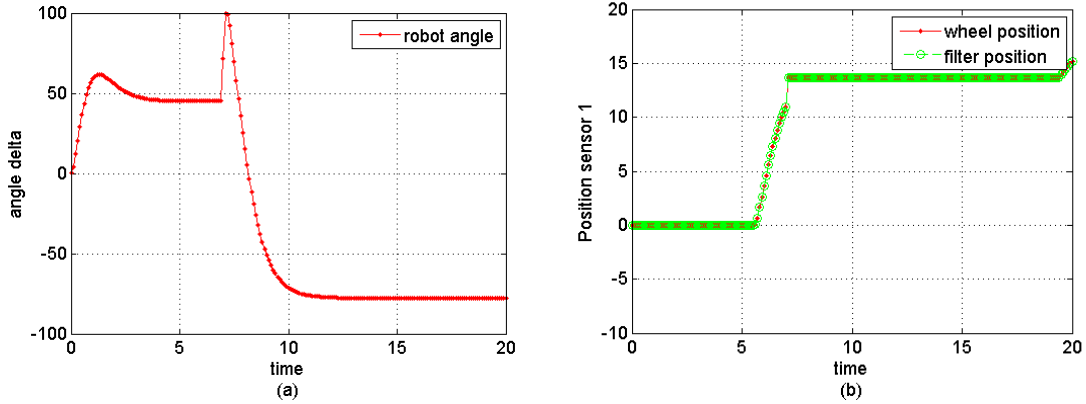


Figure 9. The Angle and Sensor 1 Position Signal

Figure 9 (a) shows that the angle rotation of the TWMR, angle against time graphs in first 20 second. The angle delta of the TWMR decline in 7.2th second due to the over running of the TWMR. Figure 9 (b) shows the front sensor range distance in first 20 second. The sensor position is always positive when the TWMR is in forward running. Therefore the error of the overshoot front side in added positively to the system.

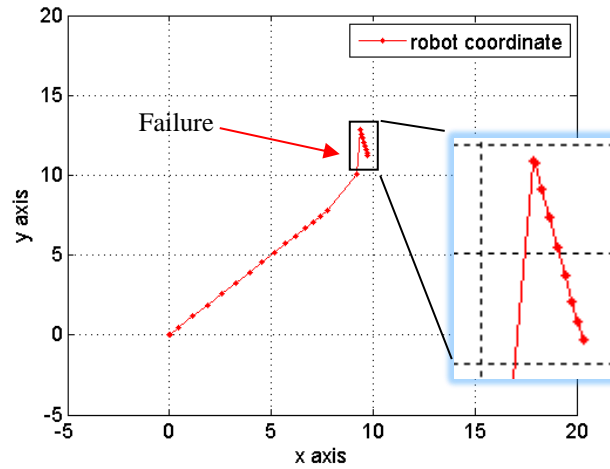


Figure 10. Map of Two Wheel Mobile Robot

Figure 10 shows the location of the robot in every 0.1 second, and the initial value started from (0, 0) and end with (10, 10). The failure showed that the TWMR are sliding to the left, resulting the faulty to the actuator 2 as positive forward, making the TWMR deviate from the desire location. In 7.2s second, TWMR will make another reference good position to the desire location in the system.

C. Simulation 3

In the third case, the non-well function in the sensor 1 was interrupted by the hardware failure where the wires lose connection, resulting a permanent fault. Therefore, in order to mimic the fault of this situation, a fault signal is added after 2nd second of the simulation time by 10 percent to the Sensor 1. In this condition, the fault replacement method (solution 4) is carried out.

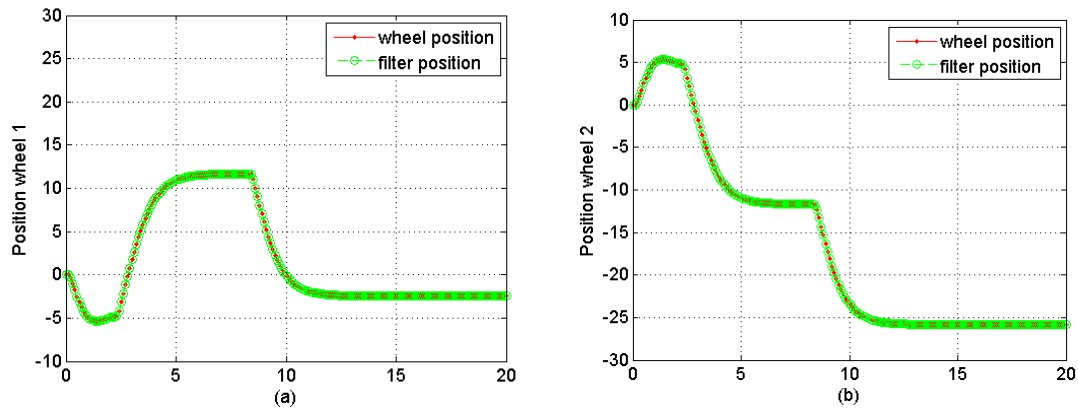


Figure 11. The Actuator 1 and Actuator 2 position graph

Figure 11 represents both the real position (dot line) and the predicted position (circle line). Figure 11 shows that there is no actuator fault in both actuators, where the errors of the wheels position and the filter position are zero. In 2.2th second, the TWMR

will operate in the fault of the sensor 1 by replacing the sensor 3 as the front side detect distance sensor and the running mode is in the reverse condition.

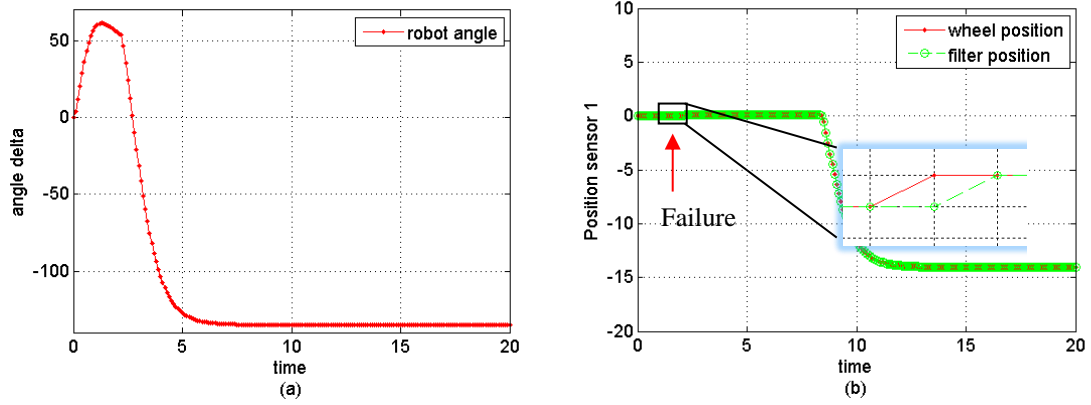


Figure 12. The Angle and Sensor 1 Position Signal

Figure 12 (a) shows that the angle rotation of the TWMR, angle against time graphs in first 20 second. The angle delta of the 2.2s second are started turn negatively and continuous running in -135° degree. Figure 12 (b) shows the front sensor range distance in first 20 second. The fault appears on t time moment $t = 7s$ and the out u_t of the sensor is 10% up of its nominal value. The sensor position signal is dropped in to negative side the TWMR is in the reverse condition.

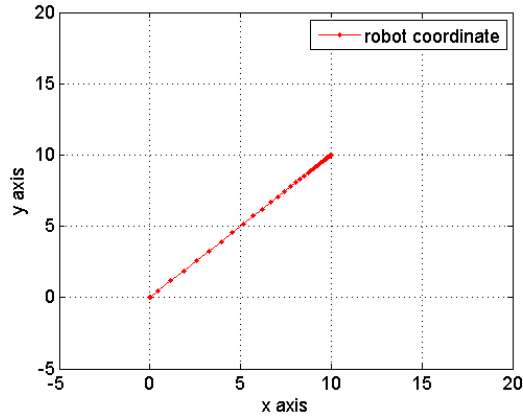


Figure 13. Map of Two Wheel Mobile Robot

Figure 13 shows the location of the robot in every 0.1 second, the initial value started from (0, 0) and end at (10, 10).

The simulation result above has been verified where it shows how the algorithm implement the solution and identify the fault of the system. The result reveals that it has successfully solved the fault during the time of fault occurrence in the system. FSM model produces location value within 2s and has instructed the robot to adjust its angle to the desired destination.

CONCLUSION

We have presented an FT algorithm to construct a set of a fault solution (subsystem) by decomposition of the fault tolerant algorithm which is intended to be integrated into real-time environment. The FT system is capable to avoid, replace, reset and recover any possible fault occurred in the system. This FSM method is also useful in multi-faults environment, giving an innovative solution to identify fault and solve fault immediately. The resolving speed is 0.2s, including giving instruction to stop running. However, some faults of the system cannot be solved perfectly such as correcting the sensor to its initial state. In addition, faults involving timing and operational statistic cannot be handled by the FSM approach. The research work is currently under simulation condition.

The future research should investigate the solvability of the proposed system on the different type of faults as well as the implementation of this method towards fault accommodation.

ACKNOWLEDGMENT

This works is supported by Ministry of Higher Education Malaysia, under the Fundamental Research Grant Scheme, FRGS RDU140137.

REFERENCES

- Apley, D. W., Shi, J., & Arbor, A. (n.d.). (1998) Diagnosis of multiple fixture faults in panel assembly, *Journal of Manufacturing Science and Engineering*, Vol. 120, No. 4, pp: 793--801
- Baghernezhad, F., & Khorasani, K. (2013). A robust fault detection scheme with an application to mobile robots by using adaptive thresholds generated with locally linear models. *2013 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, 9–16. <https://doi.org/10.1109/CICA.2013.6611657>
- Bai, L. S., Dick, R. P., Dinda, P. A., & Chou, P. H. (2011). Simplified programming of faulty sensor networks via code transformation and run-time interval computation. *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011*, 1–6. <https://doi.org/10.1109/DATE.2011.5763023>
- Benowitz, N., Calhoun, D. F., & Alderson, G. E. (1975). An Advanced Fault Isolation System for Digital Logic, *c*(5).
- Bouhouche, S., Lahreche, M., Ziani, S., & Bast, J. (2005). Fault detection and monitoring of length loop control system in pickling process. *Proceedings - International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA 2005 and International Conference on Intelligent Agents, Web Technologies and Internet, 1*(1), 642–647. Retrieved from <https://www.scopus.com/inward/record.uri?eid=2-s2.0-33847207175&partnerID=40&md5=374de3cf9046d89d97aa436892c192bd>
- Crestani, D., & Godary-Dejean, K. (2012). Fault Tolerance in Control Architectures for Mobile Robots: Fantasy or Reality? *7th National Conference on Control Architectures of Robots*. Retrieved from http://hal.archives-ouvertes.fr/docs/00/80/43/70/PDF/2012_CAR_FTRobotic-FantasyOrReality.pdf

- Dedeoglu, G., Roumeliotis, S. I., & Sukhatme, S. (2000). Fault Detection and Identification in a Mobile Robot Using Multiple Model Estimation and Neural Network, (April).
- Dugan, J. B., Bavuso, S. J., & Boyd, M. A. (1992). Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems. *IEEE Transactions on Reliability*, 41(3), 363–377. <https://doi.org/10.1109/24.159800>
- Fourlas, G. K., Karras, G. C., & Kyriakopoulos, K. J. (2015). Sensors Fault Diagnosis in Autonomous Mobile Robots Using Observer - Based Technique, 49–54.
- Kamel, M. A., Yu, X., & Zhang, Y. (2016). Design of Fault-Tolerant Cooperative Control Algorithm Applied to WMRs Against Actuator Faults, 7092–7097.
- Kanoun, K., Kaaniche, M., Baounes, C., Laprie, J. C., & Arlat, J. (1993). Reliability Growth of Fault-Tolerant Software. *IEEE Transactions on Reliability*, 42(2), 205–219. <https://doi.org/10.1109/24.229488>
- Lee, C., Alena, R. L., & Robinson, P. (2005). Migrating fault trees to Decision Trees for real time fault detection on international space station, 1–6. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1559584%5Cnhttp://ieeexplore.ieee.org/ielx5/10432/33126/01559584.pdf?tp=&arnumber=1559584&isnumber=33126
- Magni, L., Scattolini, R., & Rossi, C. (2000). E 2 < E 2 0, 30(6), 860–865.
- Mathew, J., Mathew, K., Azeez, N. A., Rajeevan, P. P., Gopakumar, K., Wang, T., ... Member, S. S. (2014). A Cascaded Multilevel Inverter Based on Switched-Capacitor for High-Frequency AC Power Distribution System. *IEEE Transactions on Power Electronics*, 27(1), 156–163. <https://doi.org/10.1109/TPEL.2013.2290313>
- Shieh, Y.-B., Ghosal, D., Chintamaneni, P. R., & Tripathi, S. K. (1989). Application of Petri net models for the evaluation of fault-tolerant techniques in distributed systems. *Distributed Computing Systems, 1989., 9th International Conference on*, 151–159. <https://doi.org/10.1109/ICDCS.1989.37943>
- Wang, C., & Schwartz, M. (1992). Fault detection with multiple observers, 1–10.
- Zhu, X., Wang, J., Qin, X., Wang, J., Liu, Z., & Demeulemeester, E. (2015). Fault-Tolerant Scheduling for Real-Time Tasks on Multiple Earth-Observation Satellites. *IEEE Transactions on Parallel and Distributed Systems*, 26(11), 3012–3026. <https://doi.org/10.1109/TPDS.2014.2363657>